### Challenges

From which source should consumers retrieve the trained model? Which data transfer method is best given the current workflow?

- Trained models can be cached at various producer locations: GPU, DRAM, local storage, and PFS.
- How can we choose and dynamically adjust the model update frequency to optimize both training and inference performance? (Figure 2)

- Frequent model updates slow down training due to continuous model checkpoint interruptions on the producer side.
- Infrquent updates may cause inference delays or inaccurate inference results as consumers use outdated models.

### Initial Evaluations

![Figure 3: Impact of model update frequency to training and inference](image)

**Figure 3:** Impact of model update frequency to training and inference.

**Table:** Initial Evaluations

<table>
<thead>
<tr>
<th>Time(s)</th>
<th>GPU</th>
<th>DRAM</th>
<th>PFS</th>
<th>FTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.12</td>
<td>3.5</td>
<td>0.33</td>
<td>1.8</td>
<td>0.5</td>
</tr>
<tr>
<td>0.31</td>
<td>3.5</td>
<td>1.24</td>
<td>2.5</td>
<td>0.85</td>
</tr>
</tbody>
</table>

### Conclusions

- This work designs and implements an I/O framework to accelerate DNN model delivery and discovery between producers and consumers (Figure 3).
- Design Objectives:
  - Cache-Aware Model Handler: Reduces end-to-end model update latency.
  - Lightweight Push-Based Notification: Minimizes model discovery latency.
  - Intelligent Performance Predictor: Predicts inference performance over a duration and dynamically adjust model update frequency.

### References


---

**Figure 1:** Coupling of training and inference within deep learning workflows.

---

**Figure 2:** Impact of model update frequency on training and inference.

---

**Figure 4:** Lightweight push-based notification.